

Template Matching

Victor Marcius Magalhães Pinto

1 Descrição da Tarefa

Usando as técnicas de template matching discutidas em sala, apresente a taxa de acerto para cada uma das 5 classes das imagens faciais disponibilizadas abaixo. Como as taxas de acerto se alteram em função da escolha de diferentes características para representar o problema?

2 Algoritmo

Conforme o enunciado da questão, a tarefa consistia em obter características de 50 fotografias, e realizar a comparação de 25 delas, de treinamento, com outras 25, de teste, a fim de verificar se as características obtidas para cada uma possibilitavam a sua correta classificação. Para tanto, foram implementadas 3 funções de features, e foram testadas as comparações com cada uma a fim de verificar quais características foram mais propícias para realizar as classificações, através de suas taxas de acerto.

2.1 Feature 1

A primeira das funções de feature tem como objetivo obter parâmetros estatísticos do documento. No caso, faz uso da média dos valores dos pixels da imagem, de seu desvio padrão e de sua soma, além da contagem de quantos pixels possuem valores dentro de intervalos espaçados de 10 em 10. A função, juntamente com a função auxiliar de contagem de cor pode ser vista a seguir.

```
> contaCor <- function(x) {  
+   x <- as.array(x)  
+  
+   co <- array(0, 10)  
+  
+   for (i in x) {  
+     if (i < 10)  
+       co[1] <- co[1] + 1  
+     else if (i < 20)  
+       co[2] <- co[2] + 1  
+     else if (i < 30)
```

```

+         co[3] <- co[3] + 1
+     else if (i < 40)
+         co[4] <- co[4] + 1
+     else if (i < 50)
+         co[5] <- co[5] + 1
+     else if (i < 60)
+         co[6] <- co[6] + 1
+     else if (i < 70)
+         co[7] <- co[7] + 1
+     else if (i < 80)
+         co[8] <- co[8] + 1
+     else if (i < 90)
+         co[9] <- co[9] + 1
+     else if (i < 100)
+         co[10] <- co[10] + 1
+ }
+ return(co)
+ }
> features1 <- function(x) {
+     m <- mean(x)
+     t <- sum(x)
+     s <- sd(x)
+
+     co <- contaCor(x)
+
+     return(c(m, t, s, co))
+ }
>

```

2.2 Feature 2

A segunda das funções de feature implementadas seleciona todos os pixels na metade superior da imagem, e todos os pixels na metade inferior, as tratando como vetores, depois realiza o cálculo de distância euclidiana entre os mesmos, juntamente com os dados de soma, média e desvio padrão dos pixels da imagem. A função de feature, juntamente com a função de distância euclidiana pode ser vista a seguir.

```

> # Função de Feature
> features2 <- function(x) {
+     d <- matrix(x, nrow = 56)

```

```

+     v1 <- d[1:as.integer(dim(d)[1] / 2), ]
+     v2 <- d[as.integer(dim(d)[1] / 2 + 1):dim(d)[1], ]
+
+     v2 <- as.vector(v2)
+     v1 <- as.vector(v1)
+
+     f <- sim(v1, v2)
+     m <- mean(x)
+     t <- sum(x)
+     s <- sd(x)
+
+     return(c(f, m, t, s))
+ }
> #Cálculo da Distância Euclidiana
> sim <- function(a, b) {
+   if (length(a) == length(b)) {
+     size <- length(a)[1]
+     ed <- 0
+     for (i in seq(size)) {
+       ed <- ed + (a[i] - b[i]) ** 2
+     }
+     ed <- sqrt(ed)
+     return (ed)
+   }
+ }
>

```

2.3 Feature 3

E finalmente, a terceira feature recebe os dados das metares superior e inferior, assim como da metade esquede e da direita, e as trata como vetores. em seguida, realiza a distância euclidiana entre as 4, a cada momento realizando comparações diferentes, e retorna como resultado o resultado dessas comparações. a função pode ser vista abaixo:

```

> features3 <- function(x) {
+   d <- matrix(x, nrow = 56)
+   v1 <- d[1:as.integer(dim(d)[1] / 2), ]
+   v2 <- d[as.integer(dim(d)[1] / 2 + 1):dim(d)[1], ]
+   v3 <- d[, 1:as.integer(dim(d)[2] / 2)]
+   v4 <- d[, as.integer(dim(d)[2] / 2 + 1):dim(d)[2]]

```

```

+
+     v4 <- as.vector(v4)
+     v3 <- as.vector(v3)
+     v2 <- as.vector(v2)
+     v1 <- as.vector(v1)
+
+     f1 <- sim(v1, v2)
+     f2 <- sim(v3, v4)
+     f3 <- sim(v1, v3)
+     f4 <- sim(v2, v4)
+     f5 <- sim(v1, v4)
+     f6 <- sim(v2, v3)
+
+     return(c(f1, f2, f3, f4, f5, f6))
+ }

```

2.4 Testes das Features

Para cada uma das features, portanto, foi comparada a taxa de acertos, realizando-se 125 testes, a fim de verificar qual das imagens em cada conjunto de 5 rostos distintos correspondia à mesma face de teste.

Para a primeira função de features, a taxa de acertos foi:

Acertos: 103

Acertos em %: 82.4 %

Para a segunda das funções de feature:

Acertos: 103

Acertos em %: 82.4 %

E para a terceira função de feature:

Acertos: 106

Acertos em %: 84.8 %

2.5 Análise

É possível perceber que avaliar dados estatísticos simples não se mostra a melhor opção para a comparação entre as imagens, apesar do fato de o melhor teste ter uma diferença de apenas 3 *matches* para o pior. A melhor comparação entre as metades de cada imagem, assim como realizado na terceira função, foi a que apresentou o melhor resultado, o que comprova que uma análise mais profunda de cada imagem, extraindo parâmetros frutos de comparações mais complexas nos fornece melhores características descritivas de uma imagem, nos levando a concluir que um método mais bem embasado de extração de features se mostra essencial para uma taxa próxima da perfeita de acertos.